PeerWise Discovery and Negotiation of Faster Paths

Cristian Lumezanu, Dave Levin, and Neil Spring
{lume,dml,nspring}@cs.umd.edu
University of Maryland, College Park

ABSTRACT

Routing overlays have the potential to circumvent Internet pathologies to construct faster or more reliable paths. We suggest that overlay routing protocols have yet to become ubiquitous because they do not incorporate mechanisms for finding and negotiating with mutually advantageous peers: nodes in the overlay that can benefit equally from each other. We show that mutually advantageous peers exist in the Internet and that one-hop detour routing is sufficient for a latency-reducing overlay. We then simulate such an overlay construction process to show that it is efficient and scalable.

1. INTRODUCTION

Routing overlays [2, 12, 22] allow end-hosts to control the path taken by a packet in the Internet. They have been proposed as a means to address the shortcomings of BGP [16], which does not explicitly select low-latency routes [17] or adapt quickly to failures [7].

Routing overlays make a diversity of paths available to applications and end users, paths that may be more reliable, less loaded, shorter, or have higher bandwidth, than those chosen by ISPs; that is, the potential benefit is high. At the same time, the cost of deploying a routing application on a workstation or intermediate proxy seems low. The high benefit at low cost raises the question, "why are routing overlays not standard practice?"

One possible explanation is that they do not scale [12]. Constant link monitoring and all-to-all probing, required for the fastest performance and best decisions, consume resources even when not needed. Further, an all-to-all overlay makes available far more paths than may be necessary for reliability and performance: most paths are useless.

Another possible explanation is that routing overlays were designed assuming cooperative, similar nodes, and that the Internet is instead heterogeneous and filled with freeloaders, adversaries, and miscreants. The heterogeneity of connectivity means that well-connected nodes are more likely to provide service, while poorly connected nodes are more likely to request service. This asymmetry discourages the well-connected nodes from joining—they have little to gain and would pay highly. The presence of freeloaders and adversaries implies that routing overlay designs should include an incentive mechanism: a means by which nodes compel each other to provide (nearly) as much service as they receive.

In this paper, we propose PeerWise, a routing overlay network in which peers negotiate and establish pairwise connections to each other based on their mutual advantage. Explicit peering based on reciprocity allows overlays to be built on self-interest rather than on altruism. As inter-domain routing benefits from the long-term reputation and SLA organization among autonomous systems, we believe that overlay routing could also gain from explicit, long-lived peerings. The condition of mutual advantageous peerings provides an upper bound on the cost that a peer pays to be part of the network: in a worst case scenario, when everybody is selfish, nobody will pay more than it benefits. Furthermore, because a node only peers with a few other nodes from which it can obtain benefit, the cost of wide scale deployment for PeerWise would be significantly reduced.

We apply PeerWise in the context of reducing path latency. Nodes that can help each other find shorter paths peer. To find the best peers automatically, each node computes its network coordinate and uses the error in the embedding (how over- or under-estimated any link latency is) to find Internet paths to be avoided or preferred.

We make the following contributions. We present a routing overlay in which peerings are constructed based on self-interest and mutual advantage and show that *one hop detours are enough* to find most potential latency reduction and that *mutually advantageous* peerings are available to avoid high-latency paths (§3). PeerWise is scalable because we exploit triangle inequality violations to select the best peerings (§4) and provides fairness because no user must pay more than it benefits. We present a novel path discovery and negotiation protocol (§5). Our simulation results show that this protocol finds the shorter paths between hosts in the Internet (§6). Lastly, we discuss future directions and the implications of applying PeerWise to today's Internet (§7).

2. RELATED WORK

Routing overlay networks [2, 17] improve the performance and robustness of packet delivery in the Internet by forwarding packets along links in self constructed meshes. Several strategies are used to determine which nodes should peer and what links should be followed. RON [2] builds a fully connected mesh and monitors aggressively all existing edges. Other approaches sacrifice unnecessary edges for scalability to define more sparse meshes: Nakao *et al.* [12, 13] employ topology information and geography-based distance prediction to build a mesh that is representative of the underlying physical network. None of these approaches has any mechanisms to ensure fairness and cooperation among nodes. We focus on finding shorter paths, and use network coordinates and embedding errors to detect which nodes are more likely to peer with each other. We achieve scalability by using an underlying network coordinate system and ensure fairness by requiring that each peering is mutually advantageous.

Our technique of predicting good peers is similar to the TIV alert mechanism described by Wang *et al.* and used to remove bad nodes in the neighbor selection mechanisms of Vivaldi [4] and Meridian [24]. However, *our focus is not to avoid the edges that are part of triangle inequality violations, but to exploit them in finding shorter detours.*

Ensuring cooperation in forwarding has been studied extensively in wireless ad hoc networks. End-to-end paths must be created from wireless hosts who selfishly wish to conserve their battery and capacity. Our focus is different (easier): we assume correctness in the form of standard Internet routing—any host can send to any other host without our system—and wish only to find a *faster* path. Wireless routing protocols must turn to money [25], trusted hardware [1], or more extreme measures [9] to achieve end-to-end incentives across potentially *many* hops. We will show that, to reduce latency between two nodes in the Internet, just *one additional hop* suffices, allowing for much simpler incentive mechanisms, such as tit-for-tat.

Various file swarming systems [3, 10, 20] apply tit-for-tat-like schemes to induce cooperation among peers. Tit-for-tat applies when there is a *mutual interest* among peers, which is common in file swarming; for any pair of peers, one may have blocks the other does not. To locate peers of mutual interest, BitTorrent uses trackers and peer discovery to determine who has what. We show that, perhaps surprisingly, mutual interest is common in low-latency routing in the Internet as well, and that locating nodes of mutual interest can be done in a decentralized fashion.

3. PEERWISE MOTIVATION AND DESIGN

We consider routing overlays in which peers negotiate and establish pairwise connections to each other based strictly on mutual advantage. This design principle is modeled on autonomous system peerings in the Internet; SLAs are negotiated *before* the peering is realized, allowing both sides to evaluate whether they wish to enter into the agreement. SLAs are maintained over long periods of time, allowing long-term reputation to motivate cooperation. We believe that bringing the negotiation and reputation to the application level would allow overlays to (1) be built on self-interest and (2) give users an opaque view of what their cost-to-benefit ratio will be before committing any of their resources.

We focus on agreements of *mutual advantage*; any pair of nodes that can benefit from each other's resources or position in the network should peer. Clearly, incentives for cooperation are simple to implement under mutually advantageous peerings; a simple tit-for-tat scheme could ensure a long-lived, fair agreement. Further, pairwise, mutually advantageous peerings provide a powerful, dynamic, fine-grained admission control mechanism. Connections are not made based on the membership to a group, but are negotiated individually by each participant with all other participants.

However, the correctness and efficiency of such an approach is not obvious; several technical questions arise: Are strictly pairwise agreements sufficient to obtain connectivity with a routing overlay, or are complex, multi-hop paths necessary? Will all nodes find mutually advantageous latency agreements, or are some nodes universally disadvantageous? In this section, we answer both of these in the affirmative, and show PeerWise's approach is feasible in obtaining low-latency, incentive-compatible overlay paths in the Internet.

Dataset	# nodes	Source	Avg. Latency Reduction: Unlimited detours	Avg. Latency Reduction: One detour or less
PL1	192	[21, 4]	12% (1.9)	10% (1.9)
PL2	384	[19]	92% (1.6)	79% (6.9)
King1	256	[5, 4]	74% (3.8)	50% (5.0)
King2	1953	[5, 4]	94% (0.7)	78% (4.6)

Table 1: Latency reduction compared to direct paths; values in parentheses represent variance.

3.1 One extra hop is enough

In discovering low-latency paths, it is tempting to allow paths of arbitrary length. However, the cost of optimal latencies is high; finding the paths would require an expensive routing protocol such as AODV [15], and ensuring cooperation across multiple hops is difficult (see §2). Gummadi *et al.* observed that relaying through a single intermediate hop could escape many network failures [6]. We present a similar result for reducing latency—that limiting paths to one intermediate hop suffices—using four latency data sets. The King data sets consist of latencies between DNS servers and the PL data sets between PlanetLab nodes. Details of the measurements can be found in the references in Table 1.

Figure 1 compares the latencies without any detours, with an unlimited number of detours, and with at most one detour, in each of the four data sets. We summarize in Table 1. Using only a single detour, similar latency improvements are possible. These values are obtained using complete knowledge of the latency matrix, and are unlikely to be obtained with a decentralized routing protocol made scalable by aggregating information. We return to this point at the end of the section.

What little is lost in latency improvement is, we believe, outweighed by the simplicity. End-to-end incentives are handled by the pairwise agreements. We show in Section 4 how to detect these shorter paths using a clever, completely decentralized protocol.

3.2 Mutual advantage is common

Each participant in an overlay network contributes its resources in exchange for the resources of others. Unfortunately, free access and unrestricted demand may lead to over-exploitation of certain resources, especially those of well-provisioned, well-connected nodes. This tragedy of the commons occurs because the benefits of utilizing common resources accrue to individuals, while the costs of exploitation are distributed among all those who provide the resource.

Pairwise peerings based on mutual advantage would be an effective means to resolve this, as nodes could freely discriminate among the connections they allow. However, such decentralized policy may be costly: if nodes accept only the

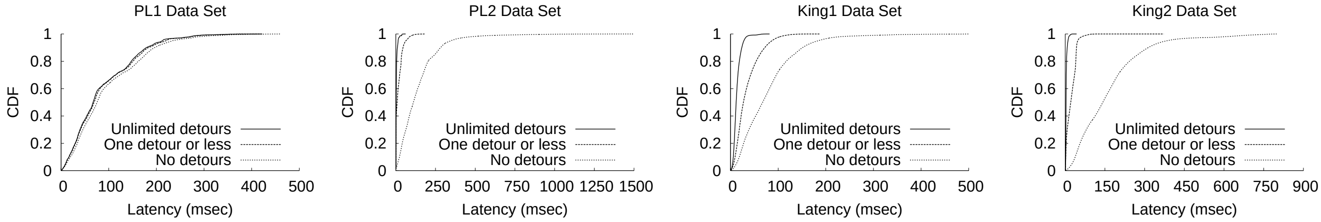

Figure 1: Allowing just a single detour achieves nearly optimal latency.

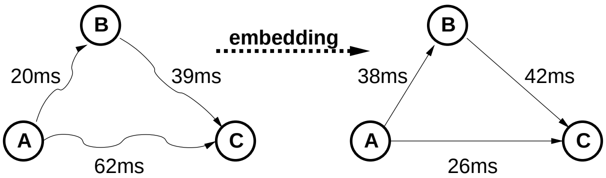

Figure 2: Embedding three points that form a bad triangle into a geometric space introduces inaccuracies.

peerings that are mutually advantageous, perhaps the benefit of the overlay is lost. For example, if those that can provide service never need it, or those who can benefit from service can never provide is, mutual advantage will not work.

Fortunately, mutual advantage is common in the latency space of the Internet. For example, as shown in Section 6, in the King1 data set, 74% of all nodes can reach each other by one-hop-detour paths discovered through pairwise peerings. This unexpected result suggests that mutual advantage is a practical criterion for pairing nodes.

3.3 Summary

The single relay hop and peerings negotiated for mutual advantage described in this section are sufficient for building a latency-reducing, incentive compatible routing overlay. The analyses in this section can be performed, however, only with the complete, global latency matrix. In the rest of this paper, we present PeerWise's scalable protocol to find low-latency one-hop detours and nodes of mutual advantage.

4. DISCOVERING SHORTER DETOURS

Because measuring and maintaining the global information required in the previous section to find detour links would be prohibitive, we look to triangle inequality violations (TIVs) in the Internet latency space to indicate alternate shorter paths between pairs of nodes. As we describe below, errors in Internet coordinates indicate the presence of TIVs and the type of error guides whether a path is a likely detour to exploit or a pathologically long link for which a detour can be found.

4.1 Triangle inequality violations

In the Internet latency space, a triangle inequality violation occurs when the latencies between a triple of nodes cannot form a valid triangle. The left side of Figure 2 presents such a scenario (ignore the rest of the figure for now). We call a triple of nodes that violates the triangle inequality a *bad triangle*. In the bad triangle ABC, AC is the *long side* and AB and BC are the *short sides*. Pairs of nodes that are long sides in bad triangles may benefit from detours; pairs that are short sides may be part of shorter detours.

Dataset	triples in TIVs	pairs in TIVs
PL1	**5%**	**66%**
PL2	**6%**	**76%**
King1	**7%**	**91%**
King2	**5%**	**92%**

Table 2: Triangle inequality violations.

Triangle inequality violations are typical, caused by Internet routing not being based on latency. ISPs may choose paths based on cost, policies, or past performance instead. We use several data sets to show that, although there may be few triangle inequality violations, many nodes can take advantage of them. For each set, we count the number of triples that form bad triangles and the number of pairs of nodes that are long sides in bad triangles (*i.e.*, pairs that have an alternate shorter path). We report the results in Table 2. Although the number of bad triangles is relatively low (less than 7%), they account for many paths not being shortest (at least 66%) These results agree with those reported by Ledlie *et al.* [8] and underline the importance of detecting and exploiting TIVs.

4.2 Network coordinates and TIVs

Network coordinates associate Internet hosts with points in a geometric space such that the distance between the points estimates the real latency between hosts. Triangle inequality violations are inconvenient for coordinate systems. Any three points that form a bad triangle cannot be embedded accurately into a space that prohibits TIVs—such as any geometric space. Inherently, the more triangle inequality violations there are, the more imprecise the embedding.

Although detrimental to the distance estimation, the inaccuracy in coordinates introduced by embedding a bad triangle can be helpful in determining which nodes and links belong to bad triangles. With this information, nodes could proactively search for shorter detours or advertise their position as relays in shorter detours for others.

When embedded into a geometric space, the sides of a bad triangle may be significantly modified (see Figure 2). The absolute error of the long side must be lower than the sum of the absolute errors of the other two sides. Therefore, we expect that the more negative the error of an edge, the higher the probability that the edge is a long side in a triangle inequality violation. Conversely, the more positive the error, the better chances for the edge to be a short side in a TIV.

The relationship between the prediction error of edges and their participation in TIVs has also been discussed by Wang *et al.* [23]. However, their goal is to avoid edges that are part of bad triangles, while we want to detect them. Further,

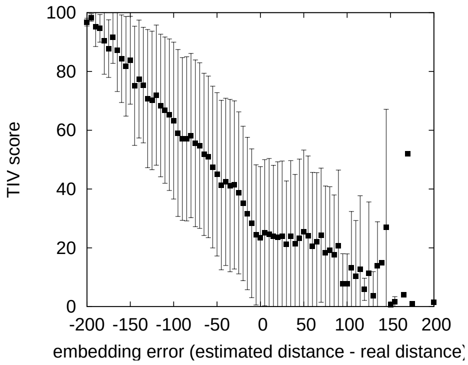

Figure 3: Average number of TIVs versus estimation error: as the estimation error decreases, it is more likely that the pair is a long side in a bad triangle

we look at absolute embedding errors of both short and long sides. Using absolute errors instead of relative errors helps us differentiate between long edges and short edges: long edges have a higher chance of being in a TIV. Finally, Wang *et al.* compute the TIV severity of pair of nodes to account for all the bad triangles in which the pair is a long side; we only need to know if two nodes form a long side in at least **one** bad triangle (*i.e.*, if there is at least one shorter path between the two nodes).

4.3 Absolute embedding errors indicate TIVs

We conjectured that as the absolute estimation error of the distance between two nodes decreases towards $-\infty$, the probability that the two nodes form a long side in a TIV (rather than a short side) increases. Similarly, as the error increases towards ∞, the two nodes will instead be the end points of a short side than of a long side in a TIV.

We use the Vivaldi network coordinate system to embed each data set into a two-dimensional Euclidean space augmented with heights. Vivaldi is distributed and adaptive, running without global state and accommodating the dynamics of the network. Our selection of embedding space draws from previous work [14, 18, 4]. In particular, Euclidean spaces have been motivated by the fact that latencies in the Internet are dominated by geographic distance and that paths generally do not "wrap around" the Earth [4].

To capture the presence of each pair of nodes in triangle inequality violations, we define the TIV score. A TIV score is given to each pair of nodes and represents the percentage of the number of times the nodes form a long side in a TIV out of the total number of times the two nodes are present in the same TIV. A TIV score of 0 means that the pair appears only as short side, while a TIV score of 100 indicates that the two nodes form only long sides in TIVs. Unlike the TIV severity metric [23], which is computed only for the bad triangles in which an edge is long side, the TIV score accounts for all bad triangles in which an edge is present.

We compute the estimation error for each pair—the difference between the embedding distance and the real distance—and plot it against the average TIV score in Figure 3. To create the plot, we took all pairs of nodes that had the same estimation error and averaged their TIV scores. The error bars correspond to one standard deviation in each direction.

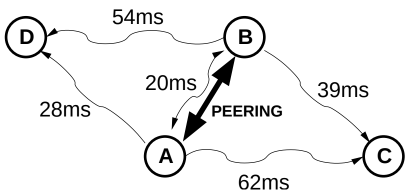

Figure 4: Example of mutually-advantageous peering: nodes A and B can help each other to find shorter paths to C, respectively D.

The outlier on the right side of the figure, corresponds to a pair of nodes for which there is only one measurement. The figure shows that as the estimation error of a pair of nodes becomes more negative, the nodes will form more and more long sides. When the estimation error becomes larger, the number of short sides that a pair forms increases. Thus, a pair of nodes with a negative estimation error has a higher chance of needing a shorter path; when the nodes have a large estimation error between them, they are more likely to be part of a shorter path for another node.

5. PEERWISE PROTOCOL

Finding shorter detours is not enough; mutual agreements between nodes are necessary. A sender node can use a detour only if the intermediate node that offers it also finds value in the sender. For example, in Figure 4, node A cannot use B to reach C more quickly unless A provides a shorter detour for B. Since the path from B to D is shorter through A, both A and B benefit from each other equally and can peer. Next we show how mutually beneficial PeerWise nodes find each other and establish peerings.

Each node maintains two tables: a *peering table* and a *negotiation table*. The peering table tracks the established mutually-advantageous peering relationships of the node. The negotiation table is an antechamber for the peering table and tracks the nodes with which no peering has been established, but which are candidates for mutually advantageous peerings. An entry in either table is associated with another node i in the system and contains i's identifier, network coordinate, and a history of round-trip times to i. The peering table adds the usage of the peering: how many times one peer uses the other as relay and the amount of data transferred in each direction. Any known PeerWise node is initially added to the negotiation table. In each negotiation session, the current node selects a peer from the negotiation table and tries to establish a mutually advantageous peering with it. If the peering is successfully established, the node is removed from the negotiation table and added to the peering table.

A new node joining the system makes a few measurements to a random set of other nodes. These node, which we call neighbors, also populate the negotiation table of the joining node. Selecting the best set of neighbors to populate the negotiation table is important for discovering more peers. We allow nodes to learn about potential neighbors from peers.

To compute their coordinates, nodes may use any distributed network embedding protocol. When its coordinates have converged, a node (the sender) can seek peerings that are mutually-advantageous. It sends a detour request to the

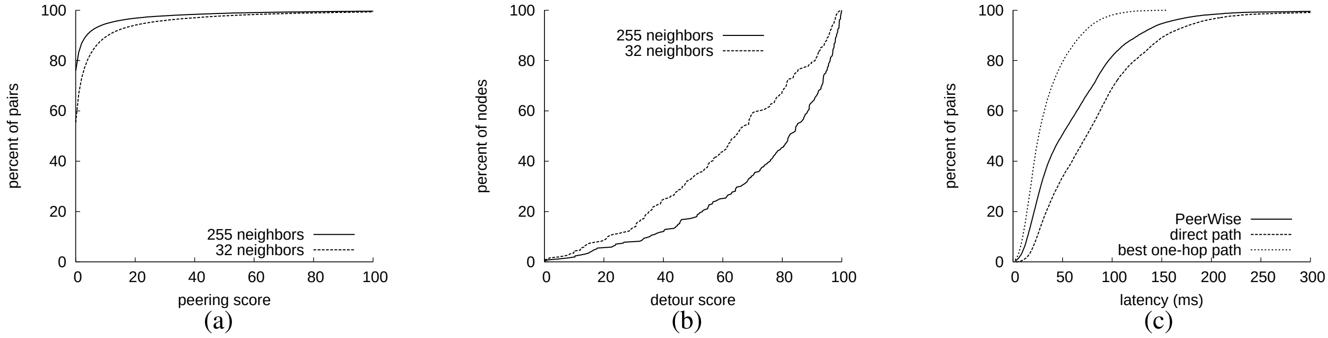

Figure 5: Cumulative distributions of (a) peering score, (b) detour score, and (c) latency

node in the negotiation table to which the distance is most overestimated and which is closer to some destinations it wants to reach. This applies the insight in Section 4: the higher the error of an edge, the more likely it is a short side in TIVs and thus part of detours. A detour request is of the form "I would like you to be my short-cut to d, and here are k nodes to which I believe I could act as a short-cut for you."

Upon receiving a detour request, a node (the receiver) can accept it, deny it, or accept it conditionally. If the receiver needs a shorter path to one of the nodes in the list offered by the sender, then a mutual advantage exists and a peering may be established between the sender and the receiver. Otherwise, if the receiver is looking for detours to destinations that do not appear in the list offered by the sender, it can pose a counter-offer, including the list of desired destinations. The sender then decides whether it can offer shorter detours to any of them. One could envision this continuing for a fixed number of rounds, or until two requests repeat.

If nodes find shorter-than-expected paths, they may advertise detours to nodes that could use these paths. Propagating good information helps the system exploit good detours and helps the node find advantageous peerings they can use later. We have not implemented detour advertisement.

The established peerings of each node are stored in the node's peerings table. The table also monitors and accounts for the traffic exchange on the peering. Nodes can establish peerings either reactively or proactively. Reactive peerings occur as responses to network conditions (*e.g.*, a node notices that the latency to a destination is greater than it should be given its coordinates), while proactive peerings ensure fast and low-cost connectivity for the future. Nodes *should* end a peering if the peering ceases to be beneficial (*e.g.*, if latencies change because of failures or congestion, making detours that are no longer shorter).

6. EXPERIMENTAL EVALUATION

Because we narrow our focus to links of mutual advantage, we do not expect our protocol to find detours to all destinations or that the detours found will be shortest. We do, however, expect it to reduce the latency to the majority of destinations. We built a prototype of PeerWise to study how well it finds detours based on partial information.

We use the King1 data set composed of 256 nodes; results for other data sets were similar. To determine network coordinates, we use the Vivaldi protocol. In our simulation, every node tries to find a shorter path to each of the other 255 nodes. Although real nodes rarely connect to all other nodes and the distribution of requests is unlikely to be uniform, we believe that forcing each node to contact every other provides a worst case for the protocol.

We define two metrics: the *peering score* and the *detour score*. The *peering score* of a pair of nodes is the difference between the number of times each node needs the other as relay after all peerings have been established. It indicates how many extra peerings (even if non-mutual) would be needed to satisfy all detour requests. To capture the performance of a single node, we use *the detour score*: the percentage of destinations that the node can reach using its mutually-advantageous peerings, out of the number of destinations for which there is a detour shorter than the direct path.

We compute the peering score of each pair of nodes and plot the cumulative distribution in Figure 5(a). When global knowledge is available, 80% of all pairs do not require extra peerings. However, when the peering table is populated with only 32 Vivaldi neighbors, around 40% of the pairs have a peering score greater than 0, indicating increased asymmetry in the peering relationships. Figure 5(b) shows the cumulative distribution of the detour score of every node. Similar to the peering score distribution, the best performance is obtained when global information is available: around 80% of the nodes can detour 60% of their destinations using the established peerings. The performance remains relatively good even if each node starts with 32 neighbors in the negotiation table: 60% of the nodes can find a shorter detour to at least 60% of the destinations to which such a detour exists. Thus, letting nodes choose peers by themselves is enough for finding most of the shorter detours.

Figure 5(c) compares the latency of the path between any two nodes using the detours discovered by PeerWise with that of the default path and of the best one-hop detour path (not necessarily mutually advantageous). Our prototype of PeerWise, with 32 initial nodes in the negotiation table can make up about half of the difference between the direct path and the best one-hop path. For the destinations to which it finds a detour, PeerWise reduces latency by an average of 25%. We expect that careful selection of the nodes in the negotiation table would lead to even greater improvement.

7. DISCUSSION

PeerWise is based on building overlay networks from *mutually advantageous* peerings; we show that such a simple, locally-enforced incentive mechanism is sufficient to provide detour routes in the Internet. Surprisingly, one-hop detour paths are enough. Equally surprising, pairs of nodes can help each other: few nodes are so well positioned that they need no help and few are so poorly positioned that they can help no one. Yet the design raises many potential questions.

Node strategies are interesting in this domain. Because we focus on wired networks, the cost of transmission and reception is low, but could become high if rate-limited upstream because of over-use (as by residential network providers). This limits how much any node should be willing to provide beyond its own usage. A node may wish to choose peerings in different directions in the Vivaldi coordinate space to gain diversity, or this diversity may emerge naturally. The direct path is likely to be an alternative to an overlay route, meaning that a node may not fear disconnection after misbehavior. Yet, to be able to use the overlay for short transactions or quickly, maintaining pre-existing peerings with good neighbors would seem in each node's interest. As further incentive, the set of "nearby", mutually advantageous nodes is relatively small compared to the size of the network, so offending such nodes may be costly. We assume that the "price" of relaying data is equal; however, a relay might believe that price is not fair and that instead, its peer should forward twice as much traffic on its behalf as it does for its peer. This "gouging" node may be correct; an auction-like mechanism may be entirely appropriate.

Extensions to other metrics are possible. Although we focus on latency as the metric to improve, it seems straightforward to apply the latency-optimized overlay toward routing around failures. Although Gummadi *et al.* [6] show that a random selection can often skirt those failures that can be avoided, the peer links chosen by PeerWise for their short latency might be too correlated to maximize resilience. For example, they may all traverse a usually high-performance but failed link. PeerWise may have an advantage in that the mutual forwarding relationship cemented over time could be more quickly applied after failures. Application to bandwidth may be more difficult, not because it would be a challenge to find pairs of nodes connected by high bandwidth, but because finding third nodes that have high bandwidth connections to one but not both of the overlay participants seems difficult. A challenging measurement study would be required to show the potential of such an overlay.

Extensions beyond pairwise connectivity are interesting. The best possible path may involve multi-hop paths; providing an incentive for a "friend of a friend" to forward is an area of future work. In general, it is NP-hard to determine if peers selfishly forming a topology will converge to a Nash equilibrium, let alone what the equilibrium is [11]. Building a topology strictly based on mutual advantage may speed convergence and ensure fairness in practice.

The dangers and effectiveness of all overlay-controlled routes, if they should become dominant, remain open to study. The "short" paths PeerWise finds may not remain uncongested when used, though it should be easy to observe poor performance along the overlay hop. Unique to PeerWise, ISP traffic engineering, if performed on a short time scale, may introduce instability to the PeerWise overlay as overused "short" paths become long to discourage their use and "long" paths become short if route choice improves. That is, the stable network constructed of stable peerings may be unraveled by rapid changes in the network.

We believe that PeerWise is a step toward practical overlay network routing in which participants can be guaranteed that their costs to participate do not outweigh the benefits.

ACKNOWLEDGMENTS

We thank Bobby Bhattacharjee and the anonymous HotNets reviewers for their helpful comments. This work was supported in part by MIPS Grant 3808 and NSF Award CNS-0626964. Dave Levin was also supported in part by a UMD CSD Dean's fellowship.

REFERENCES

[1] L. Anderegg and S. Eidenbenz. Ad hoc-VCG: A truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents. In *MobiCom*, 2003.

[2] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient overlay networks. In *SOSP*, 2001.

[3] B. Cohen. Incentives build robustness in BitTorrent. In *IPTPS*, 2003.

[4] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: a decentralized network coordinate system. In *SIGCOMM*, 2004.

[5] K. Gummadi, S. Saroiu, and S. Gribble. King: Estimating latency between arbitrary Internet end hosts. In *IMW*, 2002.

[6] K. P. Gummadi, *et al.* Improving the reliability of internet paths with one-hop source routing. In *OSDI*, 2004.

[7] C. Labovitz, A. Ahuja, A. Abose, and F. Jahanian. An experimental study of delayed Internet routing convergence. In *SIGCOMM*, 2000.

[8] J. Ledlie, P. Gardner, and M. Seltzer. Network coordinates in the wild. In *NSDI*, 2007.

[9] D. Levin. Punishment in selfish wireless networks: A game theoretic analysis. In *NetEcon*, 2006.

[10] D. Levin, R. Sherwood, and B. Bhattacharjee. Fair file swarming with FOX. In *IPTPS*, 2006.

[11] T. Moscibroda, S. Schmid, and R. Wattenhofer. On the topologies formed by selfish peers. In *PODC*, 2006.

[12] A. Nakao and L. Peterson. Scalable routing overlay networks. In *ACM SIGOPS Operating Systems Review*, 2006.

[13] A. Nakao, L. Peterson, and A. Bavier. A routing underlay for overlay networks. In *SIGCOMM*, 2003.

[14] T. S. E. Ng and H. Zhang. Predicting Internet network distance with coordinates-based approaches. In *INFOCOM*, 2002.

[15] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing. RFC 3561, IETF, 2003.

[16] Y. Rekhter and K. Lougheed. A Border Gateway Protocol 4 (BGP-4). RFC 1771, IETF, 1994.

[17] S. Savage, *et al.* Detour: A case for informed Internet routing and transport. *IEEE Micro*, 19(1):50–59, 1999.

[18] Y. Shavitt and T. Tankel. Big-bang simulation for embedding network distances in euclidean space. In *INFOCOM*, 2003.

[19] R. Sherwood and N. Spring. Touring the Internet in a TCP sidecar. In *IMC*, 2006.

[20] M. Sirivianos, J. H. Park, X. Yang, and S. Jarecki. Dandelion: Cooperative content distribution with robust incentives. In *USENIX*, 2007.

[21] J. Stribling. Planetlab all pairs ping. http://www.pdos.lcs.mit.edu/~strib/pl_app/.

[22] L. Subramanian, I. Stoica, H. Balakrishnan, and R. Katz. OverQoS: An overlay based architecture for enhancing Internet QoS. In *NSDI*, 2004.

[23] G. Wang, B. Zhang, and T. S. E. Ng. Towards network triangle inequality violation aware distributed systems. In *IMC*, 2007.

[24] B. Wong, A. Slivkins, and E. G. Sirer. Meridian: A lightweight network location service without virtual coordinates. In *SIGCOMM*, 2005.

[25] S. Zhong, J. Chen, and Y. R. Yang. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *INFOCOM*, 2003.